MERGING EVENT CATALOGS USING AGGLOMERATIVE HIERARCHICAL CLUSTERING

Jennifer E. Lewis, Sanford Ballard, Christopher J. Young, Dorthe B. Carr, Antonio I. Gonzales, and B. John Merchant

Sandia National Laboratories

Sponsored by National Nuclear Security Administration

Contract No. DE-AC04-94AL8500

ABSTRACT

Many agencies construct catalogs of the hundreds of seismic events that occur daily around the world. The Ground-Based Nuclear Explosion Monitoring Research and Development (GNEMRD) program merges these catalogs together into a composite catalog containing multiple descriptions of the same seismic event, one from each catalog of interest. The merging process requires associating seismic events in individual catalogs (herein called *origins*), that are independent estimates of the same seismic event. In this paper we describe application of classical cluster analysis techniques that provide a straightforward and robust solution to this merging problem. The resulting algorithm is much simpler to tune than the rule-based methodology used by EvLoader, which is the application currently used to merge catalogs in the GNEMRD program.

For this study, we used a simple agglomerative hierarchical clustering technique to create clusters of similar origins where the various origins in a cluster represent different estimates of the seismic parameters of the same actual seismic event. Similarity between origins is calculated using a difference measure based on latitude, longitude, depth, time, and catalog author. Uncertainty in locations is accounted for by dividing distances between origins and differences in origin time by uncertainty estimates from the catalogs. To enforce the assumption that each catalog contains only a single origin for each event, origins from the same catalog are assigned infinite difference, regardless of other parameter values.

To evaluate our new method, we processed a dataset from December 2004 to May 2005 that contains approximately 65,000 origins from 34,000 events recorded in a total of 15 different catalogs including standard global catalogs (e.g., International Data Centre [IDC], PDE) as well as various regional catalogs. We compare our results with the results generated by processing with the existing rule-based algorithm used in the EvLoader software. The vast majority of the event groupings generated by the clustering algorithm are equivalent to the event groupings created by EvLoader. Some of the event groupings that are different arise from differences in how time versus depth across origins is interpreted, and we examine these in some detail. We conclude that the new agglomerative clustering methodology produces better results and is significantly easier to tune; hence, we plan to incorporate the new algorithm into EvLoader in the near future.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
SEP 2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Merging Event Catalogs Using Agglomerative Hierarchical Clustering	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Sandia National Laboratories,PO Box 969,Livermore,CA,94551**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the 30th Monitoring Research Review: Ground-Based Nuclear Explosion Monitoring Technologies, 23-25 Sep 2008, Portsmouth, VA sponsored by the National Nuclear Security Administration (NNSA) and the Air Force Research Laboratory (AFRL)

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**9**	

OBJECTIVES

As seismic events occur around the world, different agencies construct catalogs containing information describing those events. The GNEMRD program takes these event descriptions, herein called origins, from multiple catalogs and combines them into one catalog with origins for common events grouped through an EVENT table. This process must ensure that the resulting event groupings do not contain more than one origin from the same originating catalog, and that the origins describing an event do in fact describe the same seismic event, which can be very challenging for poorly determined catalog origins.

Due to the volume of data collected on a daily basis, this process should be automated as much as possible. Most agencies that create these event groups from multiple catalogs use either an automated process or some combination of an automated process with some level of human interaction to resolve the cases the automated process was unable to handle properly.

EvLoader (Ballard and Lewis, 2004) is the current application used during integration of the multiple sets of information contained within the Air Force Technical Applications Center (AFTAC) Knowledge Base. EvLoader merges one or more events from a source EVENT table into a target EVENT table. All information linked to the source event is also merged. With EvLoader, origins in the source event can be merged with origins in the target event based on either EVID number or spatial/temporal correlation. While EvLoader yields correct results, it is essentially a rule-based application that is cumbersome to comprehend and maintain.

We propose an automated solution to this problem that uses an application of classical cluster analysis techniques, specifically a solution based on an agglomerative hierarchical clustering algorithm (Everitt et al., 2001; Romesburg, 1984). The resulting algorithm is much simpler to tune than the rule-based methodology used by EvLoader and generates clusters with errors at a rate of approximately 0.1% (errors are discussed in greater detail in the Validation section).

RESEARCH ACCOMPLISHED

Clustering is a mathematical method for determining which subsets of objects in a dataset are similar enough to be clustered together (Everitt et al., 2001; Romesburg, 1984). To cluster objects, a similarity function must be defined that returns how similar any two objects in a set are. Objects in a set that are sufficiently similar can be grouped into the same cluster.

This mechanism of assigning similarity values to pairs of origins and then grouping origins that are sufficiently similar into an event (cluster) seemed well suited to an automated process for grouping origins into events. While assorted clustering techniques were explored, an approach based on agglomerative hierarchical clustering performed well for our initial attempt at automating a solution to this merging problem. Future work will explore alternate clustering techniques (i.e., min-cut) in order to ascertain which clustering approach is the most appropriate fit for this problem domain.

Dissimilarity Function

A pivotal component to the clustering process is the mathematical function that assigns a similarity value to two objects. This function is continuous on [0, 1]. Two objects are identical if they have a similarity value of 1, and they are not at all similar if they have a similarity value of 0.

When determining how similar two origins are, the spatial and the temporal distances between two origins are used. Thus, lower distance values represent origins that are “closer” to each other (more similar) while higher distance values indicate origins that are “farther” from each other (less similar). This characteristic of how origin similarity is determined resulted in the use of a dissimilarity measure in place of a similarity measure; low dissimilarity values indicate origins that are similar while high dissimilarity values indicate origins that are not at all similar.

This dissimilarity function must assign a value used to determine if two origins have low enough dissimilarity to be contained in the same event. It must take into account the distance between the latitude and longitude of the two origins under consideration as well as their separation in time, and it must account for the uncertainties associated with these values. If two origins have the same catalog author, they are assigned infinite dissimilarity to mark them as completely dissimilar since they cannot be in the same event.

The dissimilarity function used is as follows:

$$D = \begin{cases} \infty & , \text{for } A_i = A_j \\ \sqrt{\left(\frac{\delta}{d\delta}\right)^2 + \left(\frac{t}{dt}\right)^2} & , \text{for } A_i \neq A_j \end{cases}, \qquad (1)$$

where

D is the dissimilarity of two origins (∞ = completely dissimilar, 0.0 = identical),
A_i and A_j are the catalog authors of the two origins,
δ is the horizontal separation of the two origins in degrees,
$d\delta$ is the sum of the lengths of the semi-major axes of the uncertainty ellipses surrounding the two origins, in degrees,
t is the difference between the origin times of the two origins, in seconds, corrected for origin depth as described in the Adjustments to Data section, and
dt is the sum of the origin time uncertainties of the two origins, in seconds.

Calculating the dissimilarity using this dissimilarity function results in values that are on the interval [0, ∞]. In order for these values to be useful in a clustering context, they must be scaled to values ranging from 0.0 (lowest dissimilarity) to 1.0 (completely dissimilar). Evaluation of initial results revealed that any dissimilarity value greater than 10 represented origins that could not possibly be in the same event grouping. Thus, all values generated by this dissimilarity function that were greater than 10 were simply set to a dissimilarity value of 1.0 to represent complete dissimilarity. All values less than or equal to10 were divided by 10 to produce a dissimilarity value ranging between 0.0 and 1.0.

One of the many benefits of this dissimilarity measure is how much information can be taken into account when calculating dissimilarity. Currently, the only origin data factored into this dissimilarity calculation are latitude, longitude, time, catalog author, origin time error, and semimajor axis of error ellipse information. However, if other information such as magnitude or event type needed to be utilized, this could easily be handled by modifying the dissimilarity function.

Agglomerative Hierarchical Clustering Algorithm Summary

When clustering a set of N origins, clustering begins with a set of N clusters, i.e. one for every origin. The agglomerative hierarchical clustering algorithm then assigns a dissimilarity value to each pair of origins in the set under consideration and stores these dissimilarity values in a dissimilarity matrix. For a set of five origins, the dissimilarity matrix might look something like the contents of Table 1.

Table 1. Sample Dissimilarity Matrix for Five Origins.

	1	**2**	**3**	**4**	**5**
1	0.00	0.04	0.30	0.02	0.01
2	0.04	0.00	0.09	0.07	0.40
3	0.30	0.09	0.00	0.08	0.90
4	0.02	0.07	0.08	0.00	0.15
5	0.01	0.40	0.90	0.15	0.00

The dissimilarity function returns 0.0 if two origins have no dissimilarity (are identical) and 1.0 if they have high dissimilarity (not at all similar). Recall that if two origins are from the same catalog, they are immediately assigned a maximum dissimilarity since they can never belong in the same cluster (event grouping). Notice that the matrix is symmetric because the dissimilarity of origin A with origin B is the same as origin B with origin A. Further, the diagonal terms themselves are the dissimilarities of each origin with itself (always 0.0). Thus, we only need to store the upper right or lower left portion of the matrix.

Once this matrix has been constructed, the clustering can begin. The first step in the clustering algorithm is to find the two origins that are the least dissimilar. In Table 1, this would be the dissimilarity value of 0.01 between origins 1 and 5. These origins are linked together in a cluster, and the dissimilarity matrix must be updated to contain dissimilarity information for this new cluster and the remaining origins. The dimension of the matrix (i.e., the number of rows and columns) will also be reduced by 1.

The dissimilarity between this new cluster (1,5) and the other origins (2, 3, and 4) can be computed based on the existing dissimilarity values between the origins in the original dissimilarity matrix, i.e. no new dissimilarities need to be computed. For example, when determining which dissimilarity to keep between cluster (1,5) and origin 2, the dissimilarity between origin 1 and origin 2 (0.04) and the dissimilarity between origin 5 and origin 2 (0.40) are considered. A variety of techniques exist that consider those two dissimilarities when yielding a new dissimilarity between (1,5) and origin 2 such as keeping the lowest dissimilarity, keeping the highest dissimilarity, or keeping the average. Analysis of our initial results revealed that keeping the average dissimilarity yields the best results, so the dissimilarity between (1,5) and origin 2 is set to 0.22 (the average of 0.04 and 0.40). Table 2 shows the results the resulting rebuilt dissimilarity matrix for our example.

Table 2. Rebuilt Dissimilarity Matrix after Creating the (1,5) Cluster

	2	**3**	**4**	**(1,5)**
2	0.00	0.09	0.07	0.22
3	0.09	0.00	0.08	0.60
4	0.07	0.08	0.00	0.09
(1,5)	0.22	0.60	0.09	0.00

If this process continues, every origin will eventually be combined into one all encompassing cluster, which is not the intended result. Thus, a dissimilarity threshold must be established that determines when no new clusters should be formed. This threshold is a value that represents the highest possible dissimilarity values that are allowed for origins within the same cluster. For the example in Table 2, if this threshold were set at 0.06, then all possible clusters would have been found since no remaining dissimilarity values are lower than 0.06. (Recall that lower dissimilarity values indicate higher similarity between origins.) The algorithm would have clustered five origins into four event clusters: (1,5), (2), (3), and (4). With a threshold set, this clustering process will continue to group origins into event clusters until all of the remaining dissimilarities in the dissimilarity matrix are greater than the dissimilarity threshold.

A dendrogram is a useful way to visualize which origins are grouped into clusters. The dendrogram in Figure 1 was generated from the results of clustering 77 origins into events. Events that contain only 1 origin are not represented in the dendrogram. Events containing more than one origin are represented by tree branch-like (or root-like) structures, where the lines tie together the least dissimilar two events, then the next least dissimilar event is tied to that line, and so on. The horizontal axis represents the dissimilarity value that existed between the origins when they were grouped into the same event cluster. Figure 1 demonstrates how origins with a range of dissimilarity values can still be clustered into events. Origins originally clustered into the event with EVID 29, had a higher dissimilarity value than the rest of the origins (between 0.00175 and 0.0020); this dissimilarity value was still low enough for those origins to be considered part of the same event. Origins originally clustered into the event with EVID 19 were much less dissimilar.

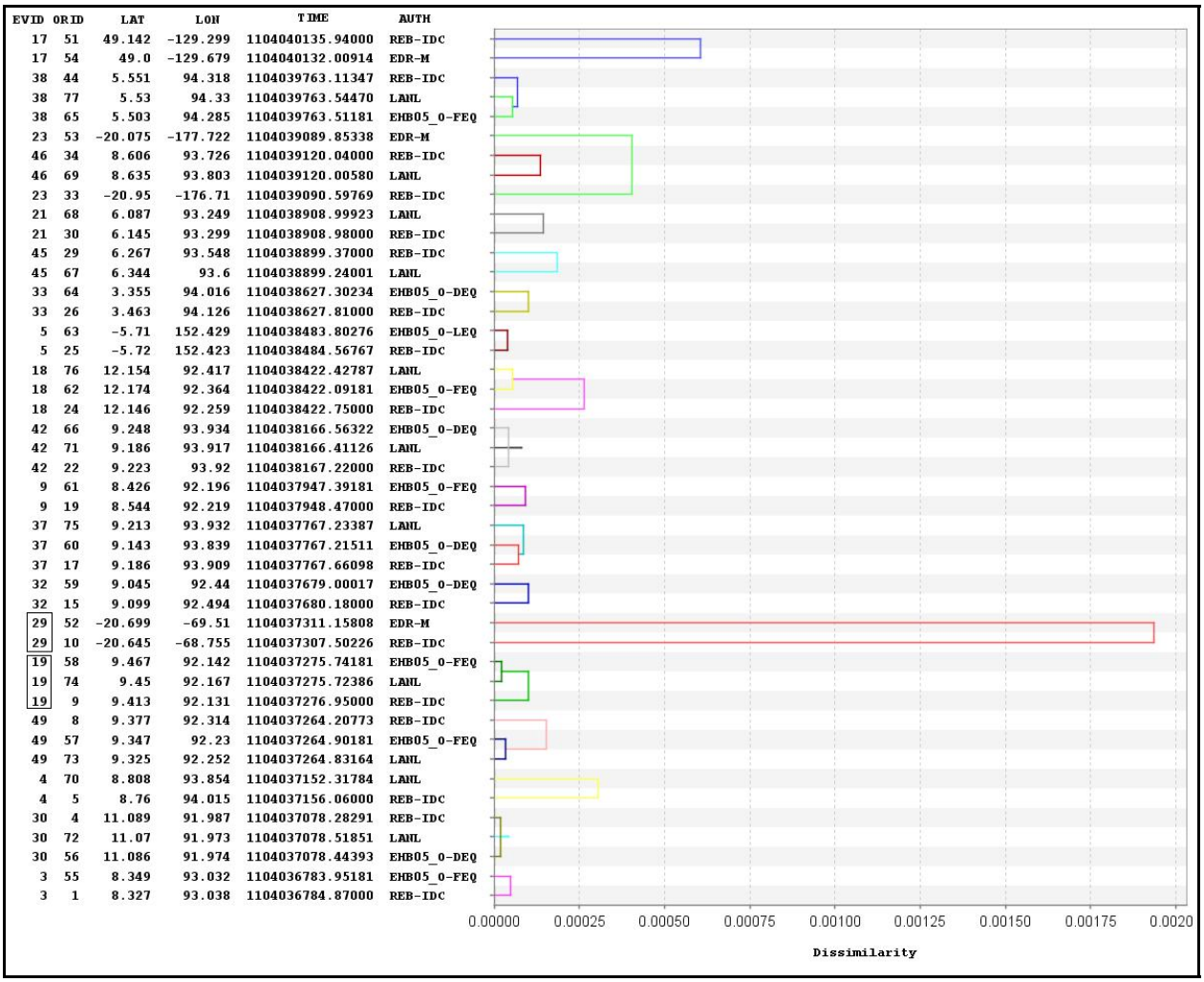

Figure 1. Sample dendrogram built from clustering 77 origins into events (only events with more than one origin shown)

Dataset Used

The origin dataset used by the clustering algorithm to generate event clusters and to check the accuracy of those clusters was a 7-month time segment of a much larger dataset collected by GNEMRD researchers at Los Alamos National Laboratory (LANL). This dataset contains predominantly global bulletin event information that LANL has collected from the following agencies:

- REB: the Reviewed Event Bulletin of the International Data Centre for the Comprehensive Nuclear-Test-Ban Treaty Organization, Vienna, Austria
- PDE: Preliminary Determination of Epicenters, the monthly version of event bulletins from the National Earthquake Information Center (NEIC)
- EHB: Engdahl, van der Hilst, and Buland which are refined origins from Engdahl et al., (1998)

In all, this dataset included origin information from 17 different catalogs, and event groupings had already been assigned and checked by LANL. Having an existing grouping provided a way to verify that the groupings generated by the clustering algorithm were indeed correct. For our test, we selected the November 2004–May 2005 timeframe

because it contains the December 26, 2004 Sumatra earthquake along with thousands of aftershocks spanning the next several months, thus providing a good test for our clustering algorithm.

This data were stored in database tables that conform to the National Nuclear Security Administration (NNSA) Core Schema (Carr, 2007).

Adjustments to Data

In order to accurately determine the dissimilarity between two origins, certain adjustments had to be made to the original data to accommodate missing uncertainty information. When no uncertainty information existed for the semimajor axis of error ellipse (SMAJAX column in the ORIGERR table), a default value of 20 km was used. When the origin time error uncertainty information (STIME column in the ORIGERR table) did not exist or was lower than a threshold of 10 seconds (an STIME threshold determined to be the best through analysis of the results of using a range of STIME values), an origin's STIME would be adjusted to be equal to the STIME threshold of 10 seconds.

Since some catalogs may have computed free-depth origins or may have fixed the origin depth at different values, we adjusted all origin times to 0 depth before calculating the dissimilarity between origins. This was accomplished by subtracting from the original origin time a correction term equal to the depth of the event, lengthened to account for an incidence angle of 30º, then multiplied by the integrated crustal slowness from the AK135 velocity model (Kennet et al., 1995).

Validation

One of the benefits of using origin data from a dataset that has already grouped those origins into events is that the event groupings generated by the clustering algorithm can easily be checked against the event groupings in the original dataset. The event groupings in the original origin data are represented in the form of the EVID column (in the ORIGIN table) that identified which event the origin belonged to, i.e., all origins in the same event have the same EVID.

Evaluation of our approach demonstrated that the number of event clusters found by the clustering algorithm was comparable to the event groupings that already existed in the LANL dataset when an appropriate dissimilarity threshold (dissimilarity values low enough to belong in the same cluster) was used as shown in Figure 2.

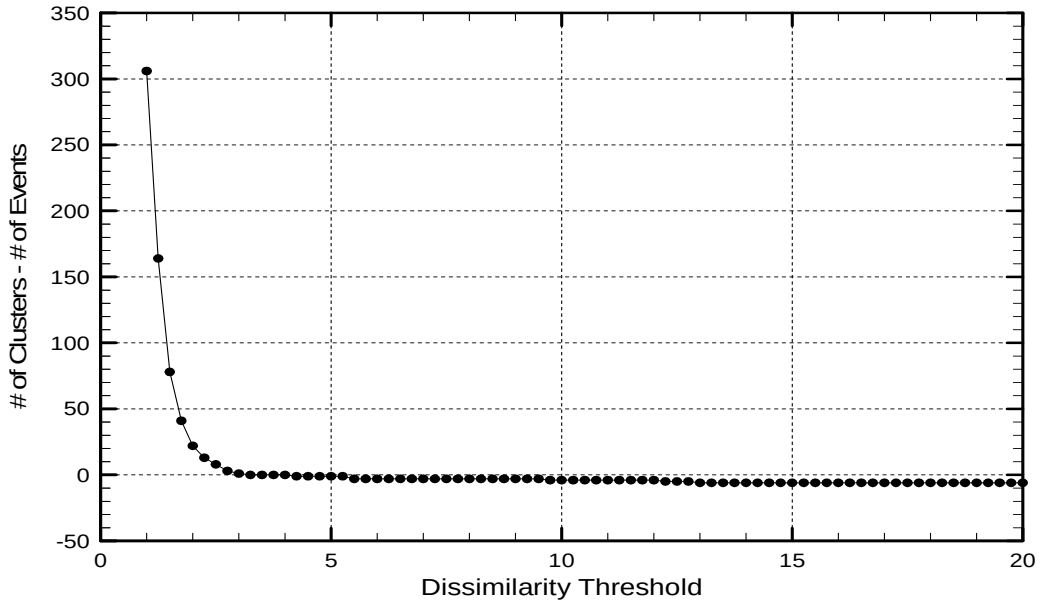

Figure 2. Difference in the number of clusters found versus the number of events designated in the original dataset as a function of the dissimilarity threshold allowed within a given cluster.

Figure 2 shows that once an appropriate dissimilarity threshold was determined, the clusters formed were comparable to the event groupings in the original data. The resulting clusters were checked to determine the following:

a) Clusters had more than one EVID in them. This would indicate that a cluster was formed that contained origins that were not originally grouped together.

b) Origins that were originally grouped in the same event were now in separate clusters. This would indicate that the algorithm determined that these origins should not be in the same event, while the original event grouping placed the origins in the same event.

Many of the errors found involved origin times that had been adjusted to depth 0; this could account for why the clustering algorithm found different event groupings than were present in the original dataset.

The clustering algorithm did not produce clustering results that were identical to the original event groupings found, although the number of generated clusters with errors was around 0.1% of the total number of clusters generated. The following types of errors were encountered: origins with low enough dissimilarity to be grouped in the same event cluster that had the same catalog author, origins that were originally in separate events that were placed in the same event cluster, and origins that were originally in the same event that were divided into separate events. The different types of errors found in the clustering results are discussed in greater detail below.

Error Scenario 1: Origins with Low Dissimilarity and Non-Unique Authors

In this scenario, multiple origins exist that have dissimilarities low enough to indicate that they could belong in the same event grouping. However, some of the origins have the same catalog author, which prevents grouping all of the origins into the same event. In some cases, the clustering algorithm did not always preserve the groupings present in the original dataset. Consider the following origin data:

	LAT	LON	DEPTH	TIME	AUTH	STIME	SMAJAX
1	-1.8306	99.5737	0	1113135202.4700	REB-IDC	0.780000 adjusted: 10.0	32.200000
2	-1.802	99.625	30	1113135207.0700 adjusted to depth 0: 1113135201.3118	EDR-M	0.380000 adjusted: 10.0	17.200000
3	-1.51	99.798	38.7	1113135202.1800 adjusted to depth 0: 1113135195.0024	EDR-M	5.250000 adjusted: 10.0	55.407895
4	-1.4674	99.8207	0	1113135196.6700	REB-IDC	1.150000 adjusted: 10.0	59.000000

The dissimilarity values for these origins are as follows (sorted from lowest to highest):

Origins 3 and 4: 0.0000956
Origins 1 and 2: 0.0001441
Origins 2 and 4: 0.0006088
Origins 1 and 3: 0.0006195
Origins 1 and 4: 1.0 (same auth)
Origins 2 and 3: 1.0 (same auth)

In the original dataset, origins 1 and 3 were in the same event and origins 2 and 4 were in the same event. However, based on the dissimilarity values, the clustering algorithm clustered together origins 1 and 2 as one event and origins 3 and 4 as another event.

Error Scenario 2: Origins Originally in Separate Events Grouped into a Single Event

The clustering algorithm would occasionally group origins that were originally in separate events into one event. Consider the following origin data:

	LAT	LON	DEPTH	TIME	AUTH	STIME	SMAJAX
1	11.2033	90.6702	0	1104059010.1300	REB-IDC	7.420000 adjusted: 10.0	63.000000
2	11.27864693	93.94496788	20	1104059031.5828 adjusted to depth 0: 1104059027.6011	LANL:GT20070601	1.135997 adjusted: 10.0	33.956238
3	11.231	93.867	20	1104059031.8200 adjusted to depth 0: 1104059027.8383	EHB05_O-FEQ	-1.000000 adjusted: 10.0	191.000000

The dissimilarity values for these origins are as follows (sorted from lowest to highest):

Origins 2 and 3: 0.0000460
Origins 1 and 3: 0.0013246
Origins 1 and 2: 0.0020136

In the original dataset, origins 2 and 3 were in the same event and origin 1 was in an event on its own. The clustering algorithm determined that these three origins should be grouped in the same event cluster since the dissimilarity values were low enough, and all of the authors were unique.

Error Scenario 3: Origins Originally in the Same Event Divided into Separate Events

In this final scenario, origins that had originally been part of the same event were separated into two different events. Consider the following origin data:

	LAT	LON	DEPTH	TIME	AUTH	STIME	SMAJAX
1	38.6051	27.4052	0	1105623537.5800	REB-IDC	1.36 adjusted: 10.0	18.5
2	39.25	27.98	10	1105623441.6000 adjusted to depth 0: 1105623439.60913	EDR-M:ATH	-1 adjusted: 10.0	-1 adjusted: 20.0

The dissimilarity value for origins 1 and 2 is 0.0053974. This dissimilarity value was higher than the dissimilarity threshold, which resulted in the clustering algorithm determining that these origins could not be part of the same event. Thus, it took two origins that had been part of the same event and separated them into two separate events.

CONCLUSIONS AND RECOMMENDATIONS

Based on the results of comparing the clusters/event groupings generated by the clustering algorithm with the event groupings already present in the LANL data and the event groupings generated by EvLoader, this clustering approach holds promise for an automated means to discovering event groupings from origin data. The high level of flexibility offered by an easy-to-customize dissimilarity function allows the clustering application to take any relevant origin information into account when determining which origins are similar enough to belong in the same event.

ACKNOWLEDGEMENTS

The authors wish to thank Julio Aguilar-Chang and Richard Stead at LANL for providing us with an excellent dataset to work with. Without such high-quality data to validate against, our job of verifying the degree to which the clustering algorithm was finding correct event groupings would have been a great deal more difficult.

REFERENCES

Ballard, S. and J. Lewis (2004). DBTools: A suite of tools for manipulating information in a relational database, in *Proceedings of the 26th Seismic Research Review: Trends in Nuclear Explosion Monitoring*, LA-UR-04-5801, Vol. 2, pp 700–709.

Carr, D. (2007). National Nuclear Security Administration knowledge base database guide, Sandia National Laboratories technical report SAND2004-0961P.

Engdahl, E. R., R. van der Hilst, and R. Buland (1998). Global teleseismic earthquake relocation with improved travel times and procedures for depth determination, *BSSA* 88: 722–743.

Everitt, B. S., S. Landau, and M. Leese (2001). *Cluster Analysis*, Fourth Edition. Oxford University Press Inc.

Kennett, B. L. N., E. R. Engdahl, and R. Buland (1995). Constraints on seismic velocities in the Earth from travel times, *Geophys. J. Int.* 122: 108–124.

Romesburg, H. Charles (1984). *Cluster Analysis for Researchers*. Lulu Press.